AUSTRALIAN OS9 NEWSLETTER

EDITOR	Gordon Bentzen	(07) 344-3881
SUB-EDITOR	Bob Devries	(07) 372-7816
TREASURER	Don Berrie	(07) 375-3236
LIBRARIAN	Jean-Pierre Jacquet	(07) 372-4675
SUPPORT	Brisbane OS9 Users Group	

Addresses for Correspondence

Editorial Material:

Gordon Bentzen
8 Odin Street
SUNNYBANK Qld 4109

Subscriptions & Library Requests:

Jean-Pierre Jacquet
27 Hampton Street
DURACK Qld 4077

Volume 4 October 1990 Number 9

AUSTRALIAN OS9 NEWSLETTER
Newsletter of the National OS9 User Group
Volume 4 Number 9

EDITOR : Gordon Bentzen
SUBEDITOR : Bob Devries
TREASURER : Don Berrie
LIBRARIAN : Jean-Pierre Jacquet
SUPPORT : Brisbane OS9 Level 2 Users Group.

Our National OS9 Usergroup lives on, thanks to the support of many subscription renewals and members who have joined us very recently. The total membership currently stands at 46, down from the 60 odd of last year, but a very healthy and encouraging number all the same.

A number of members have indicated on our membership application/renewal form that they are able to contribute articles to the newsletter. This is probably a good time to again appeal for your contribution to the content of the newsletter. We do need a much broader base of articles. Whilst we will accept material in just about any format, unformatted ASCII files on an OS9 disk is preferred. This makes it just so much easier to deal with.

We have many members who could be described as "experts" in OS9, many who are "beginners" and those anywhere between. Whatever your skill level may be, it is true that we all continue to learn from others and help others, so please assist the newsletter team by sending us something to pass on.

Now what about "good advice"? Good advice heeded is very often at far less cost than learning by experience even though lessons learned by experience are rarely forgotten. So here is a "lesson" which I have recently learned and relate in the hope that this recent experience will help others.

I added a 20 MG hard drive to my CoCo3 system earlier this year after a good deal of wondering whether it was really warranted, and I must say now that it is the most useful hardware addition I have ever made to the system. Once you have had a hard drive it is difficult to imagine how you ever did without it.

Now let's get to the good advice. "Make backups of your hard drive at regular intervals". I was given this good advice very soon after adding the H.D., and I even followed the advice with a backup at the end of each month until, yes you guessed it, a couple of months (or more) slipped by without a backup. Now it is not too hard to guess the next event in my story. A hard drive CRASH !! Now this could not happen to me - but it did.

Fortunately the term "crash" may be a little exaggeration as the only problems were a sector error, a directory entry which got screwed up somehow and claimed a couple of hundred sectors which it should not have, including a pointer to a sector occupied by the ALTBoot file. Deldir the faulty directory, right? WRONG. This simple procedure rendered ALTBoot (alternative OS9Boot file) useless. Now delete the ALTBoot and go through the procedure of putting back this bootfile from a floppy. (Yes, I did have a copy of that.) OS9 would now boot from this file. Whew! A dcheck of /H0 reported more errors than I care to mention. The H.D. is in a bit of a mess and continued use would no doubt make things worse and produce some unpredictable results.

Now is the time to "backup" all the intact files before everything is lost. After trying two different backup utilities in my possession, I find that they quit on a sector error before copying anything. HELP!!

Enter Don Berrie. After an hour or so Don had modified the bytes in a couple of H.D. sectors with his ZAP utility (published in earlier newsletters and available from our P.D. library). This eliminated the sector error and enabled me to continue with the hard drive backup.

A file backup at this point seemed the best option even though it would probably have been possible to restore the integrity of the file structure. (Well, possible if I knew what I was doing.) A problem of course was that the disk files had become very fragmented. So after the file backup, type Format /H0, go through the procedure of installing OS9Boot and ALTBoot and then restore the files from the backup set of floppies. The end result was that I lost only one directory and one file, and two night's work. Thanks Don for your help.

The moral of the story is - make backups of your floppy disks and DO maintain regular backups of the hard drive if you have one. One day you will be very glad that you did. Cheers, Gordon.

PhantomGraph printer driver changes for DMP105/6.
by Bob Devries.

One of our members has told us that he has difficulties with PhantomGraph and his DMP106 printer. The problem seems to be that the DMPTANDY.drv printer driver does not send a carriage return to the printer.

Well, I have done some detective work, and I have come up with a patch to create a DMP105_6.drv printer driver. The problem is that the printer driver is written for the newer DMP130 to 134 printers, and they use a different code for the n/144 line feed. Actually, the DMP105/6 printers only do a n/72 line feed, so I had to modify some of the code as well. Anyway, here is the information for the necessary changes. As usual, I have used MODPATCH.

Firstly, change directory to the /d1/cmds assuming that PhantomGraph is in drive /d1. Next type COPY dmptandy.drv dmp105_6.drv to make a new printer driver file available for you to modify. Next LOAD /d1/cmds/dmp105_6.drv so we can modify it in memory.

Now comes the MODPATCH code. Note that although we renamed the driver, its name in its module header is still its OLD name, so we link to it by its old name.

```
OS9:MODPATCH

1 dmptandy.drv
c 0010  74 31
c 0011  61 30
c 0012  6E 35
c 0013  64 5F
c 0014  79 36
c 0035  54 31
c 0036  41 30
c 0037  4E 35
c 0038  44 5F
c 0039  59 36
c 019F  40 5A
c 01A4  04 02
c 02CD  86 8D
c 02CE  0C E0
c 02CF  17 12
c 02D0  02 12
c 02D1  B3 12
v
```

Now, type SAVE /d1/cmds/dmp105_6.drv dmp105_6.drv

You now have a new printer driver for your DMP105 or DMP106 printer. ENJOY!!

If anyone can't do this, either because they do not have the SAVE programme, or the necessary daring, I will provide a little service for you this way: if you send your ORIGINAL PhantomGraph disk, and a BLANK OS9 formatted disk, and the return postage I will patch the drivers, and return them to you. Remember, though, I must see the ORIGINAL disk for copyright reasons.

Send them to:

Bob Devries
21 Virgo Street,
INALA. Qld. 4077

Regards,
Bob Devries.

oooooooooo0000000000oooooooooo

Basic09 Biomorphs and other mathematical mysteries.
by E.L.(Ted) Martin.

Some issues of the Scientific American magazine contain a Computer Recreations column. I was intrigued by the one on biomorphs in the July 1989 issue (p.92-95).

Biomorphs are produced by programmes similar to those used to create fractals using complex-number arithmetic. They have shapes like radiolaria and echinoids.

The Basic09 programme below, (biomorph1) reproduces a radiolarian-like form with 12 long processes.

ITERATED FUNCTION SYSTEMS

Other forms of fractals may be produced using IFS programmes. The examples are Basic09 versions of programmes from Australian Personal Computer magazine, February 1988.

EDITOR'S NOTE:

This programme requires a graphics screen to operate. To create one use the following command BEFORE running Basic09:

```
WCREATE /w1 -s=5 0 0 80 24 0 1 1
```

Then press CLEAR to go to the window and start Basic09.

```
MERGE /dd/sys/stdfonts >/w1
DISPLAY 1b 3a c8 01 >/w1
SHELL i=/w1&
```

```
PROCEDURE biomorph1
 0000      (* Algorithm by C.A.Pickover *)
 001F      (* see Scientific American, July 1989 *)
 0047      (* Basic09 program by E.L.Martin *)
 006A      DIM a,b,c,i,j,r,s:REAL
 0089      DIM h,v:INTEGER
 0094      RUN gfx2("clear")
 00A1      c:=.5
 00AC      FOR h=1 TO 100
 00BC        FOR v=1 TO 100
 00CC          r:=-10+.2*h
 00E1          i:=-10+.2*v
 00F6          FOR n=1 TO 10
 0108            s:=r*r-3*i*i
 0120            s:=r*s+c
 0130            j:=3*r*r-i*i
 0148            i:=j*i
 0154            r:=s
 015C          EXITIF ABS(r)>10 OR ABS(i)>10 OR r*r+i*i>100 THEN ENDEXIT
 018A          NEXT n
 0195          IF ABS(r)<10 OR ABS(i)<10 THEN
 01AC            RUN gfx2("point",h+250,v+50)
 01C7          ENDIF
 01C9        NEXT v
 01D4      NEXT h
 01DF      END

PROCEDURE ifs1
 0000      (* Basic implementation of random iteration algorithm *)
 0038      (* Sierpinski triangle *)
 0051      (* Australian Personal Computer, Feb. 1988, p.88 *)
 0084      (* Basic09 version by E.L.Martin *)
 00A8      DIM a(4),b(4),c(4),d(4),e(4),f(4),p(4):REAL
 00EA      DIM pk,pt,x,newx,y,newy:REAL
 0105      DIM xi,yi:INTEGER
 0110      DIM j,m,n:INTEGER
 011F      DATA 3
 0126      DATA .5,.0,.0,.5,.0,.0,.34
 015B      DATA .5,.0,.0,.5,1.,.0,.33
 0190      DATA .5,.0,.0,.5,.5,.5,.33
 01C5      READ m
 01CA      pt=.0
 01D5      FOR j:=1 TO m
 01E6        READ a(j),b(j),c(j),d(j),e(j),f(j),pk
 021B        pt:=pt+pk
 0227        p(j):=pt
 0233      NEXT j
 023E      RUN gfx2("clear")
 024B      x:=.0
```

```
0256      y:=.0
0261      FOR n:=1 TO 30000
0272        pk:=RND(0)
027B        IF pk<=p(1) THEN k:=1
0291        ELSE IF pk<=p(2) THEN k:=2
02AA          ELSE IF pk<=p(3) THEN k:=3
02C3            ELSE k:=4
02CE            ENDIF
02D0          ENDIF
02D2        ENDIF
02D4        newx:=a(k)*x+b(k)*y+e(k)
02F8        newy:=c(k)*x+d(k)*y+f(k)
031C        x:=newx
0324        y:=newy
032C        IF n>10 THEN
0338          (* PRINT n,x,y
0346          xi:=FIX(x*300+10)
0358          yi:=FIX(y*180+10)
0369          yi:=200-yi
0374          RUN gfx2("point",xi,yi)
038B        ENDIF
038D      NEXT n
0398      GET #0,n
03A1      END

PROCEDURE ifs2
0000      (* Basic implementation of random iteration algorithm *)
0038      (* Square *)
0044      (* Australian Personal Computer, Feb. 1988, p.88 *)
0077      (* Basic09 version by E.L.Martin *)
009A      DIM a(4),b(4),c(4),d(4),e(4),f(4),p(4):REAL
00DC      DIM pk,pt,x,newx,y,newy:REAL
00F7      DIM xi,yi:INTEGER
0102      DIM j,m,n:INTEGER
0111      DATA 4
0118      DATA .5,0,0,.5,0,0,.25
013D      DATA .5,0,0,.5,.5,0,.25
0166      DATA .5,0,0,.5,0,.5,.25
018F      DATA .5,0,0,.5,.5,.5,.25
01BC      READ m
01C1      pt=.0
01CC      FOR j:=1 TO m
01DD        READ a(j),b(j),c(j),d(j),e(j),f(j),pk
0212        pt:=pt+pk
021E        p(j):=pt
022A      NEXT j
0235      RUN gfx2("clear")
0242      x:=.0
024D      y:=.0
0258      FOR n:=1 TO 30000
0269        pk:=RND(0)
0272        IF pk<=p(1) THEN k:=1
0288        ELSE IF pk<=p(2) THEN k:=2
02A1          ELSE IF pk<=p(3) THEN k:=3
02BA            ELSE k:=4
02C5            ENDIF
02C7          ENDIF
```

```
02C9        ENDIF
02CB        newx:=a(k)*x+b(k)*y+e(k)
02EF        newy:=c(k)*x+d(k)*y+f(k)
0313        x:=newx
031B        y:=newy
0323        IF n>10 THEN
032F          (* PRINT n,x,y
033D          xi:=FIX(x*350+150)
034F          yi:=FIX(y*180+10)
0360          yi:=200-yi
036B          RUN gfx2("point",xi,yi)
0382        ENDIF
0384      NEXT n
038F      GET #0,n
0398      END

PROCEDURE ifs3
0000      (* Basic implementation of random iteration algorithm *)
0038      (* Fern frond *)
0048      (* Australian Personal Computer, Feb. 1988, p.88 *)
007B      (* Basic09 version by E.L.Martin *)
009E      DIM a(4),b(4),c(4),d(4),e(4),f(4),p(4):REAL
00E0      DIM pk,pt,x,newx,y,newy:REAL
00FB      DIM xi,yi:INTEGER
0106      DIM j,m,n:INTEGER
0115      DATA 4
011C      DATA .0,.0,.0,.16,.0,.0,.01
0151      DATA .2,-.26,.23,.22,.0,1.6,.07
0186      DATA -.15,.28,.26,.24,.0,.44,.07
01BB      DATA .85,.04,-.04,.85,.0,1.6,.85
01F0      READ m
01F5      pt=.0
0200      FOR j:=1 TO m
0211        READ a(j),b(j),c(j),d(j),e(j),f(j),pk
0246        pt:=pt+pk
0252        p(j):=pt
025E      NEXT j
0269      RUN gfx2("clear")
0276      x:=.0
0281      y:=.0
028C      FOR n:=1 TO 30000
029D        pk:=RND(0)
02A6        IF pk<=p(1) THEN k:=1
02BC        ELSE IF pk<=p(2) THEN k:=2
02D5          ELSE IF pk<=p(3) THEN k:=3
02EE            ELSE k:=4
02F9            ENDIF
02FB          ENDIF
02FD        ENDIF
02FF        newx:=a(k)*x+b(k)*y+e(k)
0323        newy:=c(k)*x+d(k)*y+f(k)
0347        x:=newx
034F        y:=newy
0357        IF n>10 THEN
0363          (* PRINT n,x,y
0371          xi:=FIX(x*50+320)
0383          yi:=FIX(y*18+10)
```

```
0394          yi:=200-yi
039F          RUN gfx2('point',x1,yi)
03B6        ENDIF
03B8      NEXT n
03C3      GET #0,n
03CC      END
```

oooooooooo0000000000oooooooooo

OS9 Tutorial
by Bob Montowski
edited by Bob Devries

EDITOR'S NOTE:

This tutorial was written some time ago, and is aimed at OS9 level one users. The main gist of the article is, however still relevant for OS9 level two.

This will be my first tutorial on using OS-9 and it will be for the beginners who bought OS-9 and are now ripping their hair out trying to figure out how to use it now that they have it.

OS-9 is **NOT** a programming language. It is totally different from BASIC and if you wish to program in Basic then I suggest you buy Basic09 after you are a bit more familiar with OS-9.

For people who have Disk Basic 1.0 you will need to load the OS-9 BOOT disk and RUN"*". This will then tell you to put the OS-9 Master Disk in Drive 0 and push any key to continue. If you have Disk Basic 1.1 then all you need to do is put the OS-9 Master Disk in Drive 0 and type DOS. Now that OS-9 has started up and given you your Logo and license info it will ask you for the DATE and TIME. This info is **VERY** important and should be given correctly each time you start up OS-9.

DO NOT JUST HIT ENTER, GIVE A DATE AND TIME.

This info is added to each file as it is saved to disk and will be used by OS-9 in the future to keep track of current files. The same info is also available to you to help you keep tabs on the dates and times of the files that you saved to disk. OS-9 runs on a 24 hour clock so when giving the time you must remember that times after 12 noon convert to the following:

1 pm-1300 hours
2 pm-1400 hours
3 pm-1500 hours
.
.
.
10 pm-2200 hours
11 pm-2300 hours
midnight-0000 hours

To enter Dec 25, 1985...3:30 pm you would type:

YY/MM/DD HH:MM:SS
85/12/25 15:30:00

After a date and time have been given to OS-9 you may check this time anytime you want from OS-9 by tying DATE T at OS9: prompt. If you just say DATE that is all you will get. You must say DATE T to get the date and the time. OS-9 has only a few commands already in memory. All the rest of the commands that you can use from OS-9 are on your Master Disk. Each time you give a command at the OS9: prompt the computer will check to see if the command is in memory and then it will go to the disk in drive 0 and check the /D0/CMDS directory to see if the command is in there. You must remember to type the command in correctly (SPELLING) or it won't be found when the computer goes to the /D0/CMDS directory looking for it. OS-9 can be a bit slow as it has to go to the /D0/CMDS directory each time you type a command at the OS-9 prompt but you can speed this up a bit by loading some of the commands that you will use the most in OS-9. So you could type: OS9:load dir list del attr copy You will now have the commands dir, list, del, attr, copy all in memory and they are ready for quick access.[this is already done in OS9 L2. Ed] The drawback is that they are taking up memory that you might need later. The only way around this right now is to either set your drives to run at a new faster step rate (another tutorial) or to get a Hard Disk Drive for use with OS-9. Radio Shack had OS-9 coded to run the disk drives at 30 MS. track to track and to format the

disk as 35 tracks. Both of these can be changed with a little knowledge of OS-9 or by buying some commercial software that will make the changes in OS-9 for you. Another way to speed up OS-9 is to add a 256K Ram Disk to your CoCo. With the 256K Ram board installed and the right software added to OS-9 the extra memory will act like a **very** fast 40 track disk drive. [Rammer, R0 software for L2. Ed]

OS-9 always has 2 directories that it keeps track of...One is the DATA and the other is the EXECUTION directory. When you type a command OS-9 will check the current EXECUTION directory which is /D0/CMDS at startup for the command you just typed in. When you go to do a list, dir, del, rename, etc...OS-9 looks in the current DATA directory for your file. The current DATA directory at startup is /D0. So if you just type DIR OS-9 will go and assume you meant DIR /D0. If you wish to get a directory of say the DEFS directory you must give OS-9 the whole pathlist (NAME) to the directory. In this case you would type: DIR /D0/DEFS and OS-9 will know which directory you are talking about. So how do you know what is a command? Or what is a data file? Or what is a directory? You can get this info by typing: DIR E /D0 and OS-9 will give you a directory of everything that is in the /D0 directory with exact info on each entry in that directory. You will get the date and time the entry was put on the disk and the user number (0 which means you), the entry's name, the attributes of the entry and the size of the entry in hexadecimal. It is the attributes of an entry that we will want to check. They list across like this:

```
DSPPPERW
  EWR
--------
```

That is 8 slots each of which can have a letter.

If the DIR E command shows this on a line

```
D--RW-RW
```

It would mean that it is a directory and that you and any timesharing users you had on your system could read and write to that directory. If the entry gives this back:

```
--E--ERW
```

It would mean that it is a command that can be used by you and your timesharing users and that you have the right to say copy that file, rename that file or delete that file. The timesharing user would only be able to execute the file.

If you don't want to do a DIR E on a whole disk than you can get the info you need on a single entry by typing: ATTR /D0/startup. This will printout the attributes in the same manner as the DIR E command did, but you now have the option of changing the attributes of a file on the disk. We'll use the /D0/startup file for an example

...say the ATTR /D0/startup prints this

```
------rw
```

This means that the file can be read and written to. But say you don't want to accidently delete or rename the file in the future? You can type: ATTR /D0/startup -w and the write ability to that file will be taken away. If you tried to delete that file now you would get an error message. You can use this ATTR command to change the attributes on all your important files so that they will not be deleted by accident in the future. This is kind of like having a write protect tab on your disk like in Disk Basic. But you can protect single files on the disk, or even lock out a DATA directory from having files written or deleted from it.

When I told you that OS-9 will check to see if a command is in memory and then check for it in the EXECUTION directory I left out a final thing that it does. It will go to the DATA directory and check to see if there is a DATA file there with the same name as what you typed in at the OS-9 prompt. You can check this out yourself. LIST the file startup like this: LIST /D0/startup You will see this:

```
setime </term
```

it looks like a command right? Well it is what OS-9 calls a procedure file. OS-9 will take the command you type in and first check to see if it is in memroy, if that fails it will go to the EXECUTION directory and see if the command is there, if that fails it will go to the DATA directory and see if there is a procedure file there with the name you typed in. If there is, it will read one line at a time from that file and treat it like you were typing in the lines from the keyboard. If you want to try this...Just type startup at any OS-9 prompt and the system will ask you again for the DATE and TIME to use on the system. You can build a procedure file of your own that does a little more than the startup file

does.

DO THIS at the OS-9 prompt:

OS9:build /d0/myfile

you will then see a (?) at each (?) type these lines

? dir /d0
? dir /d0/cmds
? mfree
? free
? (enter)

You will now have a data file on /D0 called myfile. If you were to type myfile at an OS-9 prompt you will then see a DIR of /D0 and then a DIR of /D0/CMDS and then you will get a mfree (memory free), and finally you will get a free (free disk space) all listed to your screen one at a time. OS-9 did all the commands in the data file as if you just typed them in at the keyboard. Not bad huh???

Now the next important thing to worry about with OS-9 is how does it keep tabs on free space in memory and on the disks??? Memory in the computer is split up in blocks of 256 bytes [Ed. 8K blocks in L2 ram]. If you do a mfree you will get back about 159 to 162 blocks of memory [Ed. in L1]. If you know that 4 blocks of 256 bytes makes one K (kilobyte) than you know you have about 40K free in memory for your programs and commands. This same idea is carried over to the disk drive. All writes to the disk are done in blocks of 256 bytes or 1 sector. A newly formatted disk will have about 640 sectors on it. But 10 of these sectors are taken away for use as directory pointers. As OS-9 only writes out to the disk in blocks of 256 bytes you will be able to get more info on an OS-9 disk than a Radio Shack Dos disk which stores data to the disk in blocks of 9 sectors (9*256=2304 bytes). Write 1 chr. to an OS-9 disk and you lose 1 sector. Write 1 chr. to a RS Dos disk and you lose 9 sectors!!!

Now do a DIR /D0/CMDS and you will see quite a long list of commands that are available to you. Don't worry about all those titles because as you learn OS-9 you will become familiar with most of them and probably not use all of them. The nice thing about OS-9 that is so different from RS Disk Basic is that it is so easy to add >MORE< commands to OS-9 than it was to add commands to the RS DOS. If you know 6809 machine language you might even write some commands that you will find useful and might want to sell or trade with other OS-9 users. If you aren't all that familiar with machine language then you can buy some new commands fo OS-9 from companies like Frank Hogg or from Computerware or D.P. Johnson. These are commands that are so easy to install on your OS-9 disk!!! All you need to do is copy them to your EXECUTION directory which is usually the /D0/CMDS directory. They are then available for your use. No worry on your part as to will they work with your OS-9 system!!! Some of these programs are actual commands that you call from OS-9 and other programs are what are called FILTERS that you pipe data thru under OS-9 (more on this in future tutorial).

And now one final thing to cover on OS-9 before I end this lesson. Is there a difference btw. upper and lower case when you type in commands??? The answer is no...no...no... If you type in DIR /D0 or dir /d0 they will both act correctly if you type LIST /D0/STARTUP or list /d0/startup they will both work correctly. OS-9 doesn't care about the case of the commands you type in. But here is a standard that you might wish to keep to, so that what is on your disk is a bit easier to understand. It is felt that if you keep all directory names in capital letters and all data/comand files in lower case you will have a better idea of what is on your disk when ;you use the dir command. I find this a useful tip and try to follow it strictly when I work with OS-9.

The next lesson in the tutorial series will be on nested directories and on pipes and filters and how they are most useful under OS-9.

Bob Montowski (International) 1-215-277-6951

oooooooooo0000000000ooooooooo

DYNACALC PATCH

In last months newsletter, we presented an interpretation of the dynacalc terminal definition file, dynacalc.trm. An infuriating (to me at least) problem with dynacalc, is the necessity to have that terminal definition file located in the current directory. I know of at least one patch to rectify this problem, and have the file located in the SYS directory on the default device, but

this patch requires that the file be renamed dtrm. That is where my problems began.

The other day, I decided to backup my hard drive, but before doing so, I decided to remove a couple of directories, and in general, do some tidying up of the whole setup.

When I came to my SYS directory, I came across the file dtrm, and did not recognize its significance. You guessed it, I deleted the damn thing. All was well, until the next time that I wanted to run dynacalc. I can tell you that that ERROR #216 that kept cropping up took some tracking down.

When I finally found the problem, I decided then and there to do something about it. The following patch is what I came up with.

You will need to change the bytes at the following offsets in DynaCalc:

OFFSET	CHANGE TO
$0B2E	$39
$0B65	$4E
$0B66	$6F
$0B67	$20
$0B68	$2F
$0B69	$64
$0B6A	$64
$0B6B	$2F
$0B6C	$53
$0B6D	$59
$0B6E	$53
$0B6F	$2F

As you may already have a patched version of Dynacalc, I have purposely not included the data from the programme as it currently stands ('cause those bytes may have been changed already). For the same reason, I have not included the CRC bytes at the end of the file. There are other patches for Dynacalc that, amongst other things, change the way that data is sent to your printer, to prevent a problem with double line feeds, and, dependant on whether you needed to do those patches, your CRC will be different to mine anyway.

That means that, after you make the patches to the module, (using debug, modpatch, ded or whatever), you will need to re-verify it.

The only other thing to do after this is to place a copy of dynacalc.trm into a directory named SYS on your default device (ie /dd). The file should be named just that, dynacalc.trm, and you may need to rename it if you have done one of the other patches.

Now, when you do a directory of your SYS directory, you will not be presented with a rather meaningless name such as dtrm, and not know what it is for.

I hope that you find this patch to be of some use. If you have any problems, please do not hesitate to contact me at the number below.

Cheers, Don Berrie (07) 375-3236.

oooooooooo0000000000oooooooooo